ASGARD: A PORTABLE ARCHITECTURE FOR MULTILINGUAL DIALOGUE SYSTEMS

Jingjing Liu, Panupong Pasupat, Scott Cyphers, Jim Glass

MIT Computer Science & Artificial Intelligence Laboratory, Cambridge, MA 02139, U.S.A.
{jingl, ppasupat, cyphers, glass}@csail.mit.edu

ABSTRACT

Spoken dialogue systems have been studied for years, yet portability is still one of the biggest challenges in terms of language extensibility, domain scalability, and platform compatibility. In this work, we investigate the portability issue from the language understanding perspective and present the Asgard architecture, a CRF-based (Conditional Random Fields) and crowd-sourcing-centered framework, which supports expert-free development of multilingual dialogue systems and seamless deployment to mobile platforms. Combinations of linguistic and statistical features are employed for multilingual semantic understanding, such as *n*-grams, tokenization and part-of-speech. English and Mandarin systems in various domains (movie, flight and restaurant) are implemented with the proposed framework and ported to mobile platforms as well, which sheds lights on large-scale speech App development.

Index Terms— Spoken dialogue systems, multilingual, portability

1. INTRODUCTION

Spoken dialogue systems (SDS) have been studied for many decades. Recently, with the popularity of speech-based applications, especially on mobile devices (e.g., Siri), SDS have been revisited extensively both by the research community and by industry. SDS covers broad research areas of speech processing and natural language processing. Currently, many off-the-shelf speech recognizers and synthesizers have become commercially available; while language processing is still a big challenge for real human-computer conversation, especially when cross-domain or multilingual applications are involved.

In the research community, multilingual SDS have been developed for many purposes ([1][3][13][14][16]), and portability has always been a major focus. For example, we have developed a web-based city guide system [8], which aggregates online restaurant reviews and provides summarized opinions to end users via spoken conversation. This system relied on an English context-free-grammar (CFG) for semantic parsing of users' spoken input [12]. To extend the system to another language, such as Mandarin Chinese, we constructed a Chinese CFG [15] to extract semantic meanings (e.g., "Cuisine: 川菜") from users' utterances, similar to the English system. To demonstrate the domain extension, we ported the system to a medical domain [7], which answers users' inquiries about side effects of prescription drugs by learning the correlations from patient-provided drug reviews. New vocabularies were added to the CFG (e.g., drugs and symptoms) and new rules were carefully constructed for semantic extraction.

Such CFG-based SDS generally have high accuracy on in-domain conversations and are relatively robust for practical use, benefiting from carefully constructed grammars and high expert control. However, the coverage of users' inputs that the systems could handle depends on manually edited grammars and rules, which are usually closed sets defined by experts. Thus, out-of-vocabulary words and recognition errors could cause problems in real human-computer conversations. Due to the heavy dependency on expert knowledge and effort, scalable system development is still a big challenge. In this sense, statistical models might be plausible alternatives in terms of scalability and portability. However, one big issue for data-driven approaches is the need for large-scale training data. The amount and the quality of training data will have a strong impact on the performance of the trained models, and therefore on dialogue performance. Furthermore, large-scale data collection requires a lot human effort and user control, which could be very expensive and time-consuming. Fortunately, nowadays, crowd-sourcing platforms (e.g., Amazon Mechanical Turk [20]) have become more and more popular, where a large pool of workers could be hired for micro-tasks. It also frees the experts from standard experimental environment control and user recruiting. Such crowd-sourcing services, if well used, could provide a promising scalable platform for data harvesting in an efficient and economical fashion.

Recently, we have developed a CRF-based movie search dialogue system [6], which was our first attempt to construct an expert-free SDS platform. In this work, we generalize the underlying framework into the Asgard architecture, where CRF [5] models are employed for sequential semantic tagging on the speech hypothesis of users' spoken utterances, and the semantic tags from CRFs are normalized for database search and response generation. To address the data issue, the Asgard framework consists of a domain/language-independent data collection platform,

which can harvest large-scale labeled data via crowd sourcing efficiently. To account for language-specific characteristics, combinations of statistical and linguistic features are employed for semantic tagging, such as tokenization, *n*-grams and part-of-speech. With the Asgard framework, we implement English and Mandarin systems in novel domains (flight, restaurant and movie) as prototypes, which are also ported to mobile devices as Apps to demonstrate platform compatibility.

2. ARCHITECTURE

The Asgard framework is designed to handle multilingual language processing in general domains. An overview of the architecture is shown in Figure 1. Each system is a stand-alone application (e.g., CityBrowser, MovieBrowser, FlightBrowser), which makes it easy to deploy on various platforms such as mobile devices. Users can access the systems via different clients (e.g., computers, tablets, smart phones), and the clients communicate with the corresponding applications on the server via WAMI [2]. In each user-system conversation, the speech hypothesis of the user's utterance will be processed through the Asgard paradigm for tokenization, semantic tagging, normalization, database search, and response generation; and the synthesized spoken response will be sent back to the user.

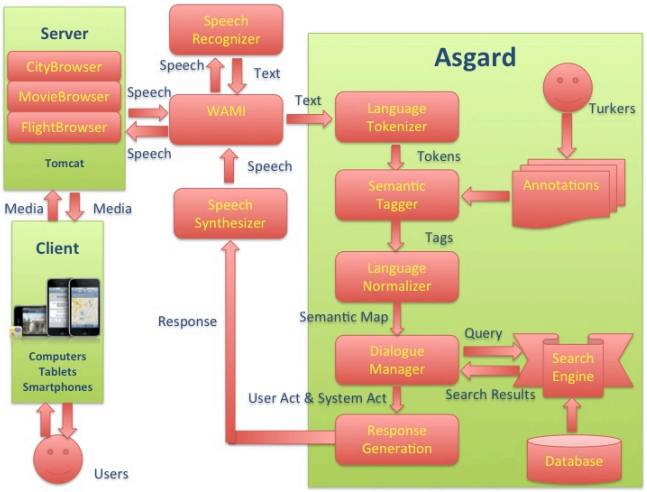

Figure 1. The Asgard architecture for multilingual SDS.

3. APPROACHES

In this section, we will explain each stage of the Asgard framework in detail, focusing on semantic tagging, data harvesting, and language tokenization and normalization.

3.1. Semantic Tagging

In the Asgard framework, we employ semi-Markov CRFs [5] for semantic tagging. Semi-Markov CRFs model the conditional probability of a segment-based label sequence given the input. More specifically, given the word sequence $x = (x_1, x_2, \ldots, x_M)$, the goal is to find $s = (s_1, s_2, \ldots, s_N)$, which denotes a segmentation of the input as well as a classification of all segments. Each segment is represented by a tuple $s_j = (u_j, v_j, y_j)$, where u_j and v_j are the start and end indices of the segment, and y_j is a class label. Segmentation and classification is jointly modeled by:

$$p(s|x) = \frac{1}{Z_\lambda(x)} \exp\left\{\sum_{j=1}^{N+1} \lambda \cdot f(s_{j-1}, s_j, x)\right\} \quad (1)$$

where $f(s_{j-1}, s_j, x)$ is a vector of feature functions defined on segments. Commonly used features include transit features, word features and lexicon features [4]. Lexicon features indicate whether a segment contains a word/phrase that belongs to an external lexicon (e.g., list of restaurant names, list of movie titles). For example, the segment-based lexicon feature is given by:

$$f(s_{j-1}, s_j, x) = \delta(s_j \epsilon\, L)\delta(y_j = b) \quad (2)$$

where L is a lexicon, b is a class, and $\delta(s_j \epsilon\, L)$ denotes that the current segment matches an element in lexicon L.

In this work, we explore combinations of statistical and linguistic features to account for language-dependent characteristics, such as segment-length (SL) features:

$$f(s_{j-1}, s_j, x) = \delta(|s_j| = k)\delta(y_j = b) \quad (3)$$

where k denotes a natural number (e.g., 1, 2, 3, …), and $|s_j|$ the length of the current segment in terms of words. So $\delta(|s_j| = k)$ represents that segment s_j contains k words.

To capture local semantic dependencies, we employ a set of *N*-gram (NG) features:

$$f(s_{j-1}, s_j, x) = \delta(\bar{x}_{s_{j-1}} = w)\delta(|s_{j-1}| = k)\delta(y_j = b) \quad (4)$$

where $\bar{x}_{s_{j-1}} = x_p, \ldots, x_q$ is the word sequence of the segment s_{j-1} preceding the current segment s_j. $\delta(\bar{x}_{s_{j-1}} = w)\delta(|s_{j-1}| = k)$ denotes that the surface string of the k words $(k = 1, 2, 3, \ldots)$ preceding the current segment is w.

Similar to *N*-gram features, we also employ a set of post-*N*-gram (PNG) features:

$$f(s_{j-1}, s_j, x) = \delta(\bar{x}_{s_{j+1}} = w)\delta(|s_{j+1}| = k)\delta(y_j = b) \quad (5)$$

where $\bar{x}_{s_{j+1}} = x_p, \ldots, x_q$ is the word sequence of the segment s_{j+1} following the current segment s_j. $\delta(\bar{x}_{s_{j+1}} = w)\delta(|s_{j+1}| = k)$ denotes that the surface string of the k words $(k = 1, 2, 3, \ldots)$ following the current segment is w.

To make use of syntactic information in the training data, we use a set of Part-of-Speech (POS) features:

$$f(s_{j-1}, s_j, x) = \delta(POS(s_j) = t)\delta(y_j = b) \quad (6)$$

where $POS(s_j)$ is the Part-of-Speech sequence of the current segment s_j, and $\delta(POS(s_j) = t)$ denotes that the POS

sequence of the current segment is t (e.g., "VB(verb)-JJ(adjective)-NN(noun)" for a three-word sequence).

Given labeled sentences, we estimate λ in (1) that maximizes the conditional likelihood of training data while regularizing model parameters. The learned model is then used to predict the label sequence s for a future input x (i.e., the speech hypothesis of a user's spoken input).

3.2 Crowd Sourcing

To make the CRF-based framework portable, there must be an easy access to training data for various applications and languages. For this purpose, the Asgard framework consists of a crowd-sourcing platform, which makes cross-domain multilingual data collection portable and scalable. We developed an AMT-based (Amazon Mechanical Turk) data harvesting platform [6], where turkers (i.e., workers on AMT) are hired to create natural language queries as well as labeling the created queries collectively. The platform supports two types of data collection: one is *frame-based* and the other is *free-style*. For the frame-based tasks, in each HIT (Human Intelligence Task), the turkers are asked to make up a natural language query based on a list of given keywords (as shown in Figure 2). The keywords will be used as the semantic labels for each generated sentence.

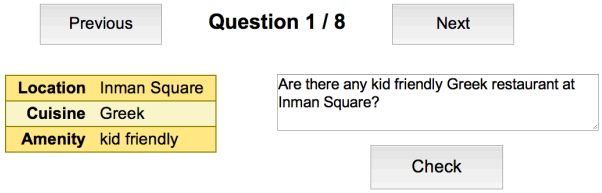

Figure 2. Screenshot of a frame-based HIT for collecting restaurant queries in English. A list of given keywords is shown on the left, and the turker is typing in a sentence on the right. The turker could click on "Check" to verify whether the sentence is legitimate or "Next/Previous" to go to another HIT.

For the free-style tasks, data is generated and annotated collectively. First, turkers are asked to create queries freely (no pre-defined keywords). A created sentence is then sent to another turker for labeling. The turker could select any segment of the sentence and choose one of the provided semantic classes as a label for each segment (as shown in Figure 3). Multiple turkers annotate each sentence and majority voting is used for aggregation. Annotated data collected by both tasks can be used for CRF model training.

3.3. Language Handler

To support multilingual systems, the Asgard framework consists of a Language Handler that unifies different language input. More specifically, a Tokenizer identifies the language of the user's input and tokenizes the utterances to a unified format, which will be subjected to the CRF-based semantic tagger. A Normalizer is then applied to the output of the semantic tagger for domain-specific query normalization. For example, "Beijing" in different languages could be normalized to "PEK", in order to maintain a universal semantic representation across languages. All language-specific and domain-specific knowledge is handled in the Normalizer. Table 1 shows an example of the normalized query for database search.

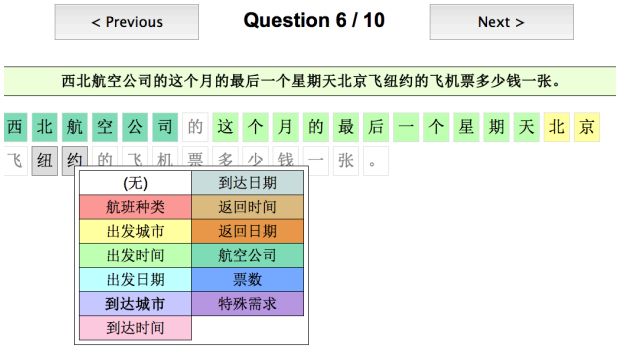

Figure 3. Screenshot of an annotation task for the flight reservation domain in Chinese. The turker selected the phrase "纽约" ("New York") and was choosing from a list of provided semantic classes (e.g., "到达城市" – "Arrival city"), which pops up every time a segment is selected. Highlighted segments are those already labeled with color-matching classes.

Table 1. Example of query normalization.

Input Utterance	**帮我订一张美国联合航空公司的下个星期四早上从波士顿到北京的单程机票**
Semantic Tags	Itinerary-type: 单程; Departure-date: 下个星期四; Departure-time: 早上; Airline: 美国联合航空公司; Departure-city: 波士顿; Arrival-city: 北京
Normalized Query	Itinerary-type: ONE-WAY; Departure-date: DEC06; Departure-time: AM; Airline: UA; Departure-city: BOS; Arrival-city: PEK

4. EXPERIMENTS

To demonstrate the portability of the proposed framework, we implemented prototype systems in multiple domains (restaurant and flight) in both English and Mandarin. Annotated data were collected for each domain and each language via the crowd-sourcing platform. Table 2 shows the semantic classes defined in each domain for annotation (both English and Chinese). We also collected more data in the movie domain for further evaluation. Table 3 shows the statistics of the experimental datasets.

For each dataset, we randomly selected 80% as the training set and the remaining 20% as the test set. Table 4 shows the semantic tagging performance with semi-Markov CRF models [11] in terms of F-score [10] (harmonic mean of precision and recall) on different feature sets (*BSL*: baseline; *LX*: lexicon features; *SL*: segment length features; *POS*: part-of-speech features; *NG*: *N*-gram features; *PNG*: post-*N*-gram features). The baseline is the combination of word features and transit features.

Table 2. Semantic classes defined in each domain.

Domain	Semantic classes
Flight	General city, General date, General time, Departure city, Departure date, Departure time, Arrival city, Arrival date, Arrival time, Return date, Return time, Transit city, Airline
Restaurant	Goal, Restaurant name, Amenity, Cuisine, Dish, Hours, Location, Price, Rating
Movie	Title, Viewers' rating, Year, Genre, Director, MPAA rating, Plot, Actor, Trailer, Song, Review, Character

Table 3. Number of sentences in each dataset.

	Restaurant		Flight		Movie
	English	Chinese	English	Chinese	English
Frame-based	8000	2180	3800	5600	6800
Free-style	8500	10100	14500	2500	5200
Total	16500	12280	18300	8100	12000

Table 4. Experimental results on semantic tagging (F-score).

Features	Restaurant		Flight		Movie
	English	Chinese	English	Chinese	English
BSL	82.87	82.91	80.85	83.52	85.84
LX	84.57	83.55	80.78	83.76	87.00
LX+SL	84.28	84.71	79.80	83.24	87.48
LX+POS	84.27	83.54	80.35	82.93	86.93
LX+NG	84.26	85.17	82.53	83.18	87.15
LX+NG+PNG	84.31	85.31	82.09	83.26	87.51
LX+NG+POS	82.74	84.95	82.29	82.98	87.51
LX+SL+NG	85.07	85.56	81.71	83.04	88.00
LX+SL+NG+PNG	84.70	85.90	81.81	83.11	88.30
LX+NG+PNG+POS	83.96	84.75	82.10	83.38	87.40
LX+SL+NG+POS	84.93	85.95	80.74	82.80	88.21
LX+SL+NG+PNG+POS	84.65	85.89	81.45	82.68	88.58

The most significant improvement was in the movie domain (88.58% vs. 85.84% of baseline), the English restaurant domain (85.07% vs. 82.87% of baseline), and the Chinese restaurant domain (85.95% vs. 82.91% of baseline). The English flight domain achieved some outperformance with lexicon and *N*-grams features (82.53% vs. 80.85% of baseline). There was not much improvement on the Chinese Flight data. One possible reason is the lack of data, especially free-style sentences (2500 compared to ~10k in other sets). Another observation is that in Chinese there are many different expressions on time and date (e.g., "三十一号之后的星期三晚上", "国庆节之前的最后一个礼拜五中午以前"), which might be difficult for segmentation. Also, turkers often found the annotation between time and date confusing (e.g., "departure_time" and "departure_date" are often mislabeled), which happened in the English flight annotation set as well. Thus, some quality control on crowdsourcing (e.g., pre-task qualification or training for workers) might help harvest cleaner data and improve the performance.

The trained CRF models were then embedded in each prototype system for language understanding. Asgard supports a seamless interface for plug-in search engines. For example, we used Lucene [17] for MovieBrowser to support multi-field retrieval on an IMDB database as well as pronunciation search (i.e., metaphone search). In the restaurant domain, location is important, as the landmarks are displayed on a map. Hence, MongoDB [18] was employed for geographical search on pre-collected restaurant databases. For FlightBrowser, we used an API from ITA [19] for flight information search. We also deployed the systems to mobile devices, and Figure 4 shows the screenshots of the systems on smart phones. For further system evaluation, we will deploy the systems on AMT as mobile Apps to collect real dialogue data from general users.

Figure 4. Screenshots of the prototype systems (from left to right: flight, restaurant and movie) on mobile devices (the interface was enlarged for clearer display). Users can click on the microphone icon on the screen to talk to the systems, or click on either the "ABC" or the "中文" button at the bottom for language switching.

5. CONCLUSIONS

In this paper, we presented the Asgard architecture, which was designed to support portable spoken dialogue systems across various languages, domains and platforms. The CRF-based framework depends on a population of non-expert workers via crowd sourcing for training data collection and annotation. Multilingual systems (English and Mandarin) in three domains (movie, flight, and restaurant) have been implemented as demonstrations, all based on the proposed framework. Domain-dependent or language-specific changes are minimized across applications. The easy deployment to mobile platforms also demonstrates possibility of large-scale speech-based App development.

For future work, we will explore the extension of the framework to other languages such as Arabic. Data filtering and quality control approaches will also be investigated for high-quality data collection via crowd sourcing.

6. ACKNOWLEDGEMENT

This research is supported by Quanta Computers, Inc. through the Qmulus project. Thanks to Ian McGraw and Dennis Smiley for the deployment of the prototype systems to mobile devices. And thanks to Stephanie Seneff and Victor Zue for helpful discussions.

7. REFERENCES

[1] J. G. Amores, G. Pérez, and P. Manchón. MIMUS: A Multimodal and Multilingual Dialogue System for the Home Domain. In Proc. of ACL, 2007.

[2] A. Gruenstein, I. McGraw, and I. Badr. The WAMI Toolkit for Developing, Deploying, and Evaluating Web-Accessible Multimodal Interfaces. In Proc. of ICMI, 2008.

[3] M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M.Walker, S. Whittaker, and P. Maloor. MATCH: An Architecture for Multimodal Dialogue Systems. In Proc. of ACL, 2002.

[4] X. Li. Understanding the Semantic Structure of Noun Phrase Queries. In Proc. of ACL, 2010.

[5] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proc. of ICML, 2001.

[6] J. Liu, S. Cyphers, P. Pasupat, I. Mcgraw, and J. Glass. A Conversational Movie Search System Based on Conditional Random Fields. In Proc. of Interspeech, 2012.

[7] J. Liu and S. Seneff. A Dialogue System for Accessing Drug Reviews. In Proc. of ASRU, 2011.

[8] J. Liu, S. Seneff, and V. Zue. Dialogue-Oriented Review Summary Generation for Spoken Dialogue Recommendation Systems. In Proc. of NAACL-HLT, 2010.

[9] J. Liu, Y. Xu, S. Seneff, and V. Zue. CityBrowser II: A Multimodal Restaurant Guide in Mandarin. In Proc. of ISCSLP, 2008

[10] C. J. Van Rijsbergen. Information Retrieval (2nd). Butterworth. 1979.

[11] S. Sarawagi and W. W. Cohen. Semi-Markov Conditional Random Fields for Information Extraction. In Advances in Neural Information Processing Systems (NIPS), 2004.

[12] S. Seneff. TINA: A Natural Language System for Spoken Language Applications. Computational Linguistics, Vol. 18, No. 1, pp. 61-86, 1992.

[13] C. Wang, S. Cyphers, X. Mou, J. Polifroni, S. Seneff, J. Yi and V. Zue. Muxing: A Telephone-Access Mandarin Conversational System. In Proc. of ICSLP, 2000.

[14] C. Wang and S. Seneff. High-Quality Speech Translation in the Flight Domain. In Proc. of Interspeech, 2006.

[15] Y. Xu, J. Liu, and S. Seneff. Mandarin Language Understanding in Dialogue Context. In Proc. of ISCSLP, 2008.

[16] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington. JUPITER: A Telephone-Based Conversational Interface for Weather Information. In IEEE Transactions on Speech and Audio Processing, Vol. 8, No. 1, 2000.

[17] http://lucene.apache.org

[18] http://www.mongodb.org

[19] http://www.itasoftware.com/index.html

[20] https://www.mturk.com/mturk/welcome